

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

# UCD

G3/82      Unclass  
26146

2425262728293031  
MAY 1977  
RECEIVED  
POSTAL FACILITY  
NEW YORK BRANCH

**College of Engineering**  
**Department of Electrical Engineering**

FINAL REPORT  
Development of a Character, Line  
and Point Display  
for  
NASA Johnson Space Center  
CONTRACT NAS 9-14696



FINAL REPORT

DEVELOPMENT OF A CHARACTER, LINE AND POINT DISPLAY SYSTEM

FOR

NASA JOHNSON SPACE CENTER

HOUSTON, TEXAS

CONTRACT NAS 9-14696

Principal Investigator -

Earle W. Owen  
Associate Professor

Contributors -

Craig A. James  
Prem C. Jain

Department of Electrical Engineering

University of California, Davis

Davis, CA 95616

MARCH, 1977

## TABLE OF CONTENTS

Introduction .....	1
The Electronic Tablet .....	4
The Display and Character Generator .....	7
The Microcomputer .....	11
The Refresh Memory .....	13
The Display Interface .....	13
The Software .....	16
Suggestions for Further Work .....	22
References .....	24
Appendix .....	25

## LIST OF FIGURES

1. The major components of the terminal .....	3
2. Layout of the electronic tablet .....	5
3. The refresh memory and point display .....	7
4. The character matrix and character space .....	8
5. The Motorola MCM6571A character generator symbols .....	9
6. The TV raster display timing .....	10
7. The microcomputer, SBC 80/10 block diagram .....	11
8. The refresh memory .....	13
9. The display interface .....	15
10. The software flow diagram .....	19

## INTRODUCTION

The system built under this contract is a compact graphics terminal intended for use as the input to a computerized medical records system. The principal mode of communication between the terminal and the records system is by means of checklists and menu selection. However, the terminal accepts short, handwritten messages as well as conventional alphanumeric input.

The advantages of automatic storage and retrieval of medical information has been evident for some time, but attempts to develop such a system have been only partially successful. Medical information is so diverse and extensive that it resists complete classification. In addition, the critical user is the physician. Physicians have limited resources of time and energy and must carefully invest these resources in many demands. A computerized system must be compatible to their habits and needs, as well as effective, if it is to be accepted.

Checklists are a convenient mode of computer entry that is consistent with present medical record keeping. However, in the past, checklist systems have failed because of the impossibility of providing for all contingencies. The UCD-NASA terminal overcomes this deficiency by allowing short handwritten notes to be entered. There is no intention, at this moment, of making these notes machine readable. The intention is to label them with machine readable characters and then store and retrieve them by means of a computerized filing system.

The ultimate goal in the development of a user compatible terminal is a portable terminal that can be carried by the user much like a notepad. Early in the development of the terminal it was realized that this

ideal would not be attained in a single stage. Instead it was decided to build a versatile, easily modified device that could be used to experiment with the novel concepts inherent in the proposed record keeping system.

As Figure 1 shows, the UCD-NASA terminal consists of several major parts: an electronic tablet, a display, a microcomputer controller, a character generator, and a refresh memory for the display.

The input device is a Talos, Inc. electronic tablet. Part of the tablet is used for an alphanumeric touch typewriter while the other part is available for checklists and handwritten messages.

The display device is a raster scan TV monitor. Alphanumeric characters are entered into a fixed character matrix with the aid of a cursor just as in conventional computer terminals. Handwritten and graphical information are entered anywhere on the screen and in any order. Both handwritten and alphanumeric characters are displayed at the same time.

The electronic system that controls the flow of information in the terminal is an Intel SBC 80/10 microcomputer. Another Intel product, a 16 kilobyte memory, stores the point-by-point array of information to be displayed. A specially designed interface continuously generates the raster display without the intervention of the microcomputer.

The UCD-NASA terminal is the first step toward developing a user-compatible computer input. A number of possible directions for further development are suggested at the end of the report.

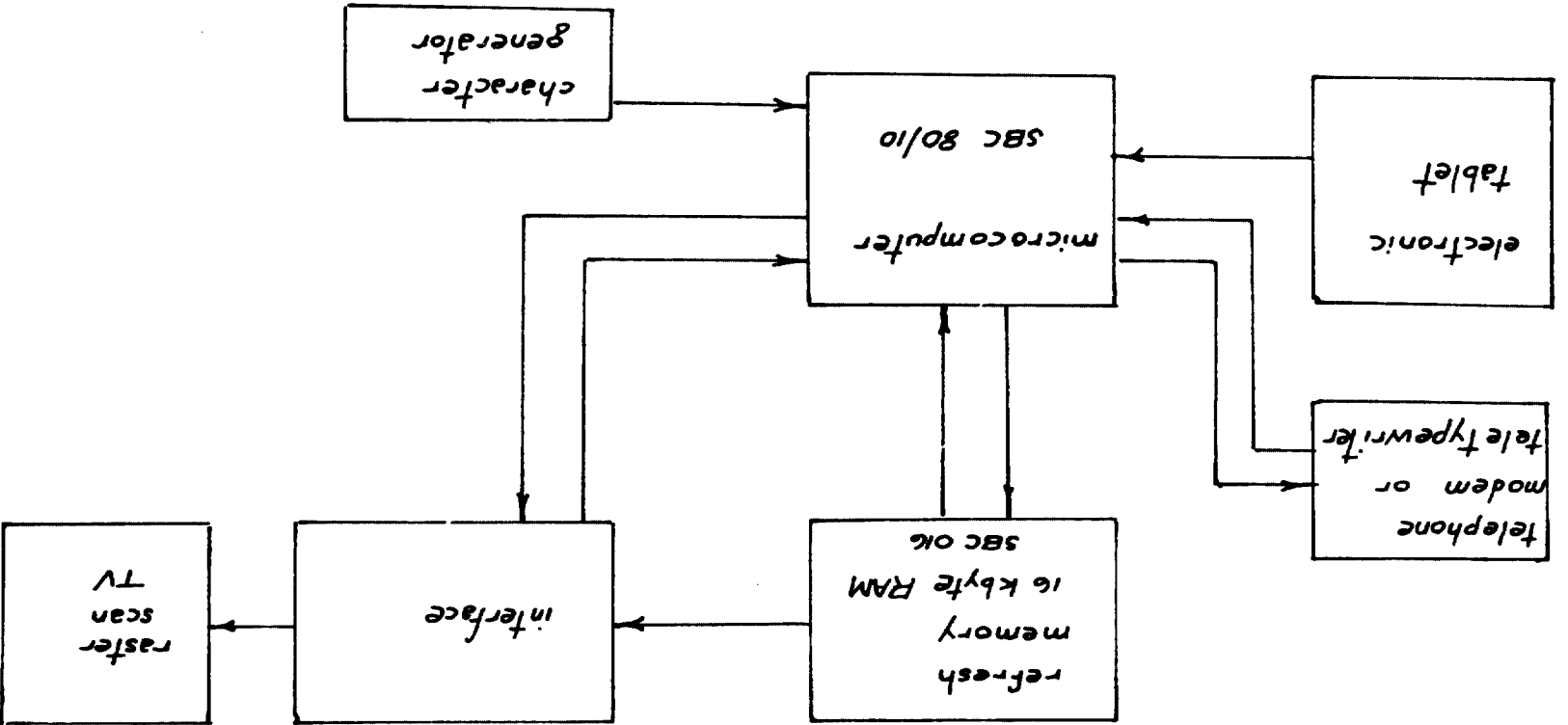


Figure 1

The major components of the terminal



## THE ELECTRONIC TABLET

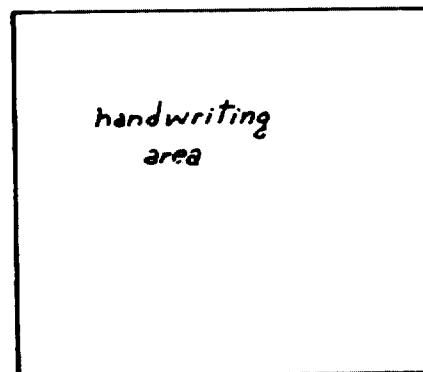
The Talos electronic tablet is an electric field device with a capacitive pickup in the form of a ballpoint pen. The model used has an active area of 11 by 11 inches, a resolution of about 64 lines per inch, and a sampling rate of up to 100 samples per second.

The surface of the tablet is divided into three areas, as shown in Figure 2. The left portion of the tablet is a touch typewriter containing the upper and lower case characters, punctuation and other symbols, a cursor control, and several other control functions. The remainder of the tablet can be used for checklist inputs. Handwriting is confined to an area approximately four inches wide by three inches high.

The tablet is operated under the control of the microcomputer in either the POINT or TRACK mode of operation. While the tablet is computer controlled the operator cannot use the mode selector switch on the tablet; the mode is selected automatically. When the pen is touched to the typewriter area the microcomputer selects the POINT mode of operation and each touch of the pen causes a single pair of coordinates to be generated. When the pen is in the handwriting area the tablet is instructed to operate in the TRACK mode, and a continuous stream of coordinates are generated at a sample rate determined by the sample rate dial on the tablet electronics case.

The touch typewriter is divided into 128 spaces, each approximately one half inch square. The majority of these spaces contain the upper and lower case letters, the numbers, punctuation, and other symbols. The top section of the tablet contains the control functions. Not all spaces are in use, a number are reserved for future software development.

COMPUTER				LOCAL			Monitor
CHART				Clear Page		Roll ↓   ↑	
GRAPHIC				Erase Point	Home	↓	Next Line
CHARACTER				Erase Line	←	Repeat	→
						↓	
•	ˆ	?	!	:	;	"	=
+	-	*	÷	(	)	[	]
<	>	*	\$	%	&	/	\
A	B	C	D	E	F	G	H
a	b	c	d	e	f	g	h
I	J	K	L	M	N	O	P
i	j	k	l	m	n	o	p
Q	R	S	T	U	V	W	X
q	r	s	t	u	v	w	x
Y	Z	0	1	2	3	4	5
y	z	6	7	8	9	SPACE	



65% of Full scale

Figure 2  
Layout of the electronic tablet

The alphanumeric characters are positioned on the display by means of cursor controls on the keyboard section of the electronic tablet. NEW LINE, ERASE LINE, and RETURN to the origin are additional character controls. Individual characters can be erased by positioning the cursor and touching BLANK. At present, editing of alphanumeric text is not possible, however, this capability can be added at a later date.

The refresh memory can store information that fills two and two thirds display screens. The ROLL control allows any part of the refresh memory to be displayed. Touching this control causes the display to roll upward or downward. The software is written so that the refresh memory appears like a loop; continual pressing of the ROLL control causes the contents of the refresh memory to be scanned repeatedly.

When the pen is moved in the handwriting space the tablet generates a stream of coordinates, which cause an image of the writing to be generated. A mixture of alphanumeric and graphical information can be displayed. The operator selects the mode by moving the pen to the appropriate area.

Touching the pen to the ERASE control before entering the handwriting space converts the pen into an eraser. As the pen is moved, an area of the display surrounding the pen is erased. The ERASE command erases whatever is displayed, graphical and alphanumeric information alike. The pen is restored to a writing instrument by touching it to ERASE for a second time.

The entire right side of the tablet, including the handwriting area, is available for charts and checklists. At present, the development of software for this purpose is at a primitive stage. However, potentially the use of this area is limited only by the imagination of the programmer.

### THE DISPLAY AND THE CHARACTER GENERATOR

The display consists of a dot matrix of 192 rows and 256 columns. While operating in the graphical mode any one of these points is accessible. As Figure 3 shows, the display occupies only part of the refresh memory. However, any part of the refresh memory can be viewed on the TV screen by means of the ROLL control.

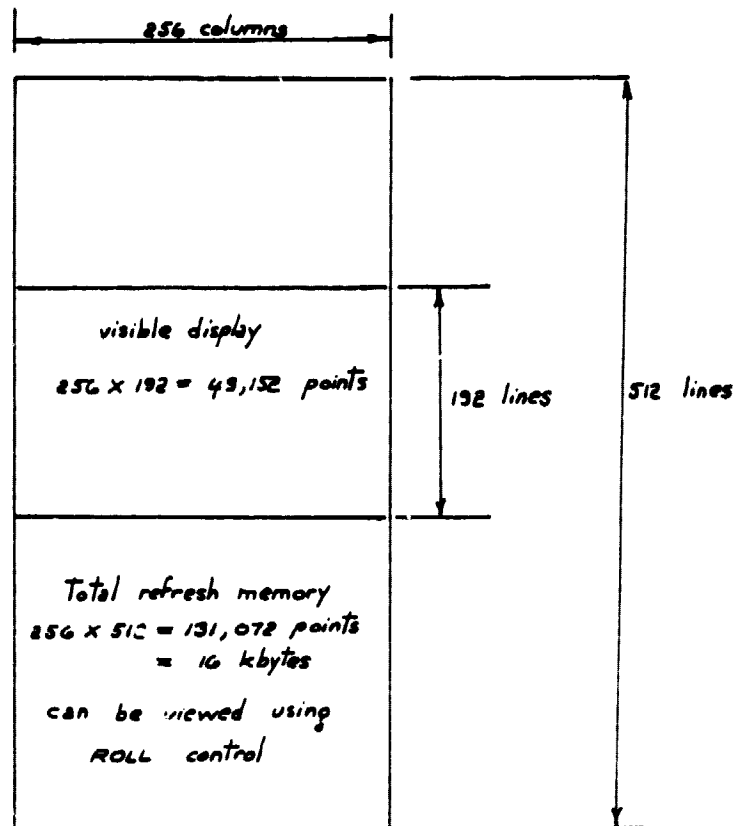
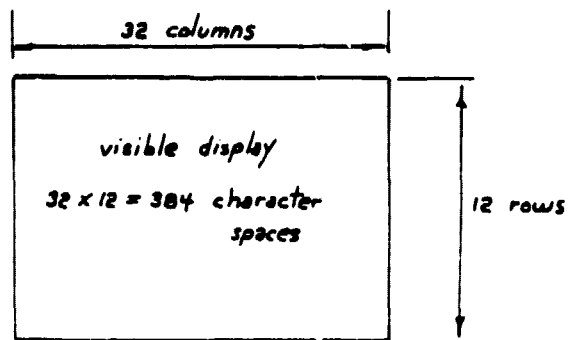


Figure 3

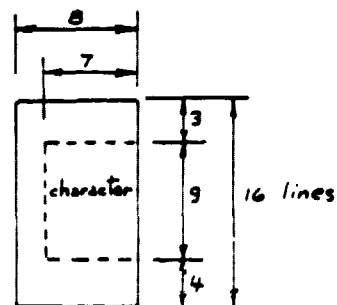
*The refresh memory and point display*

Characters originating from the touch typewriter are displayed in a matrix of 12 character lines or rows, each 32 characters long. Another 20 lines are hidden in the refresh memory. Each character space, as shown in Figure 4(b) is a rectangle eight points wide and sixteen points high. A character itself occupies a space of seven by nine points. The remainder of the character space is used for spacing, and for the cursor, which is a double line at the bottom of the character space.



(a) The visible character display

There are 20 additional character rows in the refresh memory.



(b) The layout of a character space

The character matrix and the character space

Figure 4

The dot pattern for the raster display is generated by a character generator, the Motorola MCM6571A. Figure 5 shows the characters produced by this generator and their corresponding ASCII codes. The characters that can currently be displayed are those shown in Figure 2. However, the touch typewriter can easily be redesigned and the software modified to include any of the characters in the generator.

0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Figure 5

The Motorola MCM6571A character generator symbols

The TV raster display is synchronized with the microcomputer clock. The timing and format of the raster scan, which is shown in Figure 6, is compatible with standard TV monitors. Two video outputs are supplied: a composite video signal and a vertical and horizontal drive suitable for monitors with separate vertical and horizontal inputs

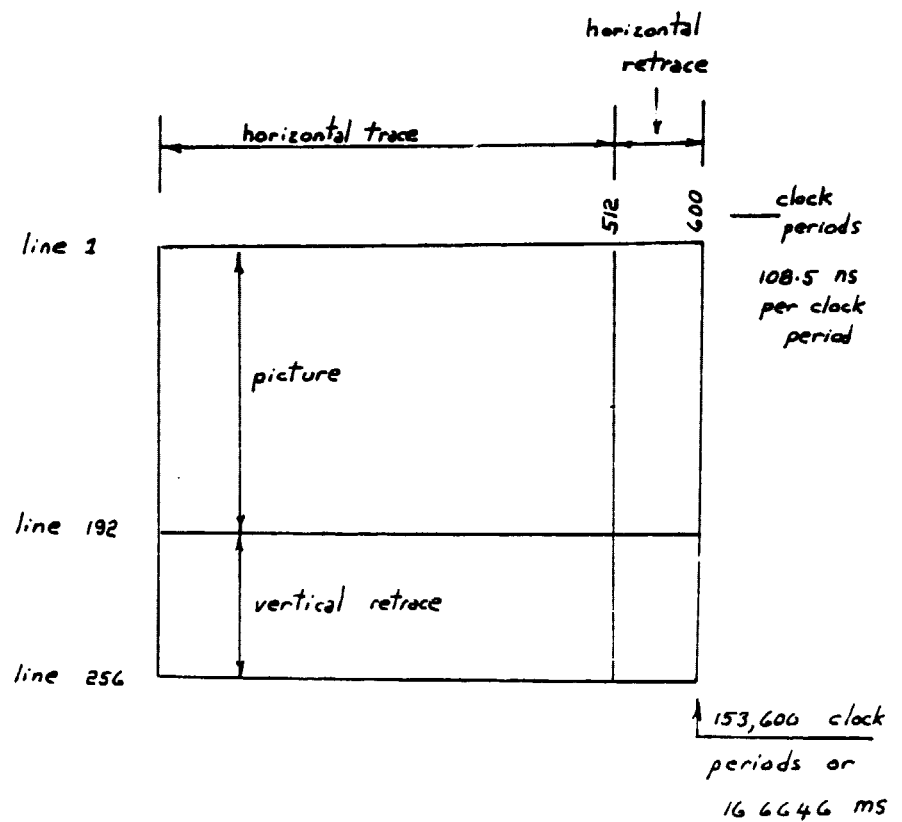
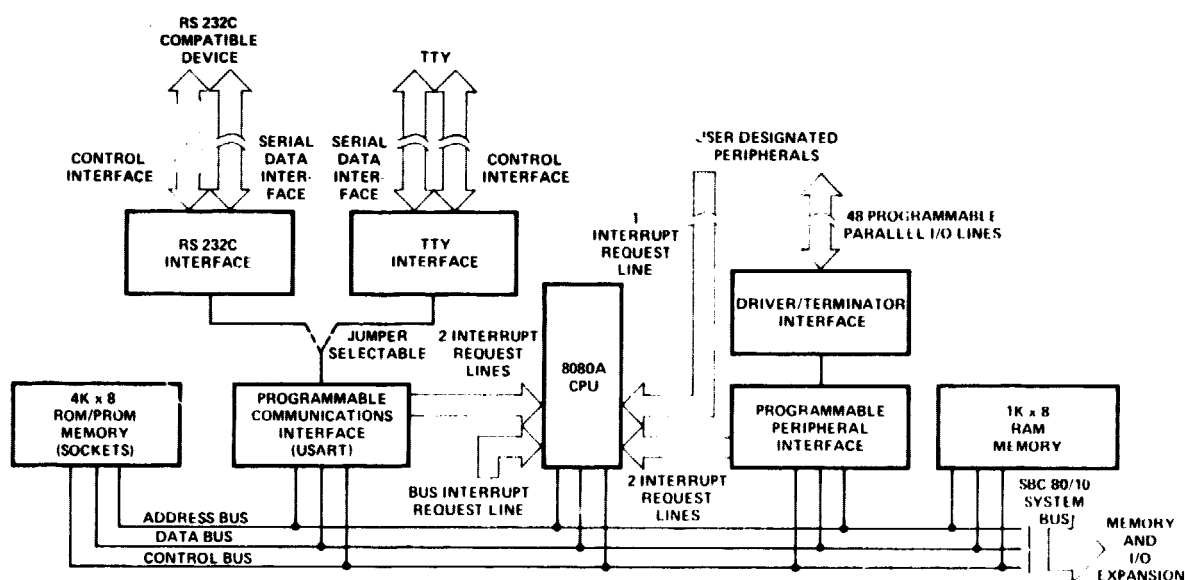


Figure 6

The TV raster display timing

## THE MICROCOMPUTER

An Intel SBC 80/10 microcomputer controls the flow of information to and from the electronic tablet, the communication port, the character generator, refresh memory, and display interface. The microcomputer was purchased as part of an Intel SBC 80P Prototype Package. This system includes a firmware monitor and a cardcage. A concise description of the SBC 80/10 and the SBC 80P is included in the Appendix. The following remarks are for the most part confined to a description of how the available options have been employed



1. Interrupts originating from the Programmable Communications Interface and Programmable Peripheral Interface are jumper selectable.

### SBC 80/10 BLOCK DIAGRAM

*Figure 7*  
*The microcomputer*

The SBC 80/10 microcomputer is a complete computer on a single board. As the simplified block diagram in Figure 7 shows, the board contains an eight bit Intel 8080A microprocessor, along with clocks, buffers, PROM, RAM, and input/output ports. There are four, one



kilobyte PROMs. Two of these, two kilobytes, house the Intel monitor, a package of system software. The other two kilobytes of PROM contain programs developed specifically for the terminal.

The microcomputer board also has available one kilobyte of RAM. This memory is used for temporary storage during data handling. Portions of it are also available for the storage of checklist programs.

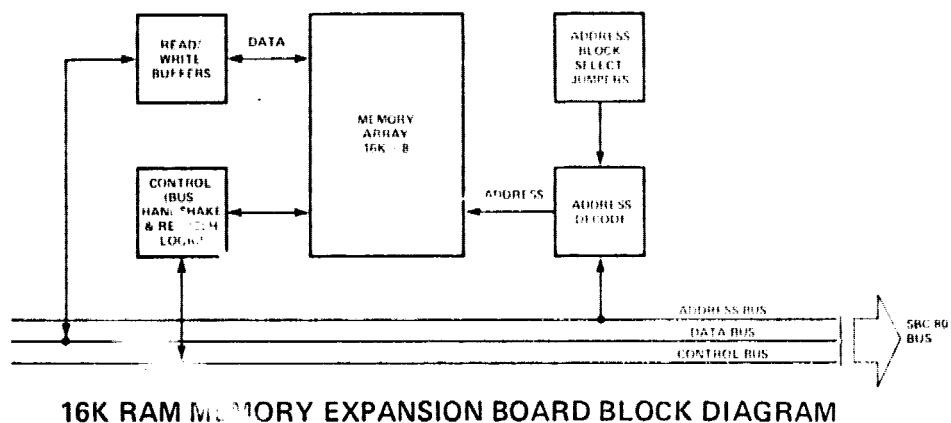
Two Intel 8255 input/output programmable interfaces, with a total of 48 lines, are housed on the board. These parallel lines are used to communicate with the electronic tablet. Each coordinate axis of the electronic tablet requires ten input lines. In addition, there are eight control lines to and from the tablet. One line is required for a computer interrupt.

The board also contains a standard RS232C interface and a teletypewriter interface. Only one of these options can be used at a time. A change from one to the other requires moving several jumpers on the board.

## THE REFRESH MEMORY

The TV display is refreshed from an Intel SBC 016, sixteen kilo-byte RAM memory board. This board is electrically and mechanically compatible with the SBC 80P Prototype Package. A concise description of its specification is given in the Appendix.

As Figure 8 shows, the memory is arranged so that information is read in and out in eight bit bytes. If a single bit is to be changed the microcomputer reads out the byte containing the bit, changes the bit and restores the byte to memory with the remaining bits unchanged.



*Figure 8*  
*The refresh memory*

## THE DISPLAY INTERFACE

The TV display interface extracts information from the refresh memory and prepares it for the raster display. The interface, which was designed and built at UCD, consists of approximately 70 integrated circuits on a board that fits into the same cardcage as the microcomputer

and refresh memory. The three boards, interface, microcomputer, and refresh memory share the same address, control, and data buses. The timing of the interface is derived from the microcomputer clock so that the interface operates in synchronism with the computer.

Using the same buses for the microcomputer, refresh memory, and interface creates a difficulty. When the interface is extracting information from the refresh memory the microcomputer cannot use the buses and therefore must remain inactive. In order to decrease this wasted time a method was devised that enables the interface to obtain data from the refresh memory in a minimum time. A line of the display is read out of the refresh memory with high speed circuits and stored in a 256 bit buffer memory. Two buffer memories are used so that while one is supplying the raster, the other is loaded from the refresh memory.

The alternative to this method is to read out the information from the refresh memory as needed. If this is done, the only time the buses are available to the microprocessor is during the horizontal and vertical retrace times. This time is about 20 percent of the total time. The use of buffer memories and fast circuits increases the time available to the microcomputer to more than 50 percent, a proportion of time that is more than adequate for the work of the microcomputer.

The flow of data within the interface, and to and from the microcomputer and refresh memory, is shown in Figure 9. The two 256 bit RAMS, each holding a line of the display, act alternately to supply the raster or to receive and store data from the refresh memory. The control, which consists primarily of a complex of counters driven by the microcomputer clock, time both the raster display and the acquisition of data from the refresh memory.

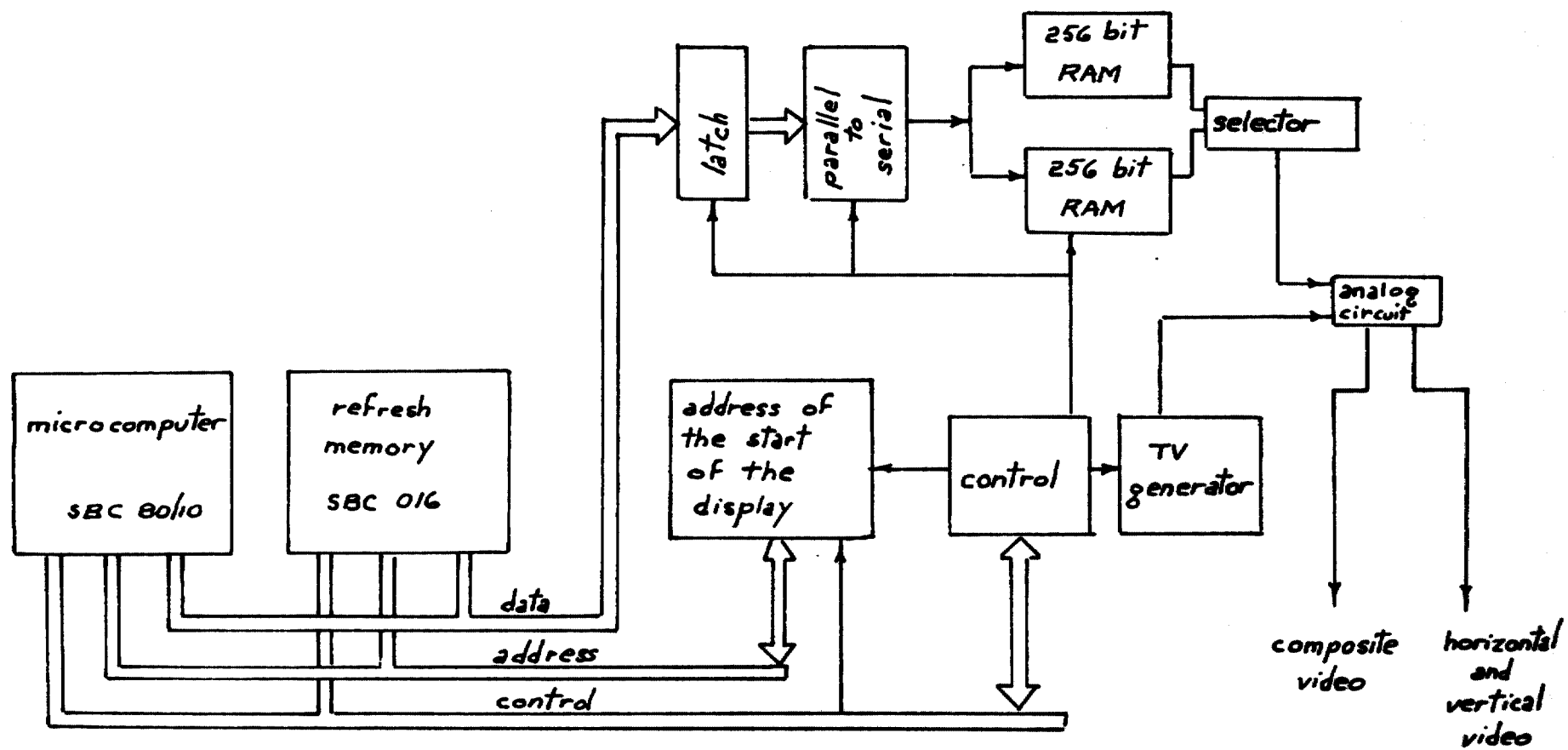


Figure 9

The TV display interface

The address of the start of the display is stored in a register in the interface. The contents of this register are changed by the microcomputer when the ROLL command is touched on the electronic tablet.

In order to save hardware, the data is transferred from the microcomputer to the address register by means of the address bus. This is done by using two of the address lines in combination with a memory read signal to indicate that the address line contains information for the address register.

The transfer of a display line from the refresh memory to the 256 RAM buffer memory is made difficult by the regeneration of information in the refresh memory itself. The refresh memory consists of dynamic RAMs that must be regenerated or refreshed approximately once every 12 microseconds. The regenerative circuits, which are part of the SBC 016 memory board, have their own clock so that the regenerative timing is not synchronized with the remainder of the system. Since the reading of a line requires more than 12 microseconds, reading of the data is interrupted at least once by a regeneration cycle. During the regeneration cycle the read operation is suspended, to begin again when regeneration is complete. Since the regeneration is asynchronous with the remainder of the system the interruption occurs at different times within the read cycle. However, in all other respects the interface, refresh read, and the generation of the raster display are locked to the microcomputer clock.

#### THE SOFTWARE

The operation of the terminal is controlled by programs stored in PROMs on the microcomputer board. These programs, consisting of approximately 500 assembly language instructions, occupy 1600 bytes of memory.

The present software does not exploit the full potential of the terminal. It was written to achieve the limited objective of operating the terminal in the stand-alone mode with the electronic tablet as the primary input device. While operating in this mode graphical and alphanumeric input from the electronic tablet are placed in the refresh memory and displayed on the cathode ray tube. Control functions such as ROLL and ERASE are provided. There is also a limited provision for operation in the checklist mode, using checklist information stored in the RAM of the microcomputer. An important adjunct to the software written explicitly for the terminal is a system software package, the system monitor, provided with the SBC 80P prototype package. The system monitor places the operator in control of the microcomputer and allows for a teletype and paper tape input.

Notably absent from the software provided with the terminal are programs for the operation of the terminal under the control of or in conjunction with a central computer. This software is yet to be developed. In addition, the existing programs for the manipulation of data from the electronic tablet can be extended and improved.

The software package is organized into a large number of sub-routines, each performing a small definable function. This organization is efficient since many of the subroutines are used in several different procedures. It also lends itself to future software development since new programs can be formed by assembling a sequence of sub-routines. The existing software structure of the terminal is easily modified by dismantling the present arrangement of subroutines and reassembling them into a new configuration.

The diagram in Figure 10 shows the flow of information within the existing programs. When the power is turned on the computer initializes itself and generates a sign on message: "UCD - NASA TERMINAL". This message is stored as ASCII codes in the PROM, along with the other software programs and information. Each character is converted into a point array by means of the character generator and stored in the refresh memory.

After "sign on" the computer goes into a halt or waiting state. When a task is complete the computer always returns to this state, remaining there until a new task is brought to its attention by means of an interrupt signal.

Since the terminal is programmed only for stand-alone operation an interrupt occurs only when there is an input from the electronic tablet. When the tablet generates an output, the computer examines the vertical and horizontal input lines to determine whether the pen is in the keyboard, graphic, or checklist area. The program then branches to the appropriate sequence of subroutines.

If the pen is in the graphic area the computer puts the tablet in the TRACK mode. While in the TRACK mode the tablet generates a continuous sequence of coordinate pairs at the sampling rate selected by a dial on the tablet. A pair of coordinates on the tablet is translated by the computer into a corresponding address in the refresh memory. To write the bit into the refresh memory the computer must first extract the byte containing the bit from the memory. The appropriate bit is then changed and the byte is written back into the memory. After having processed a pair of coordinates the computer returns to the halt state, awaiting the arrival of the next sample from the electronic tablet.

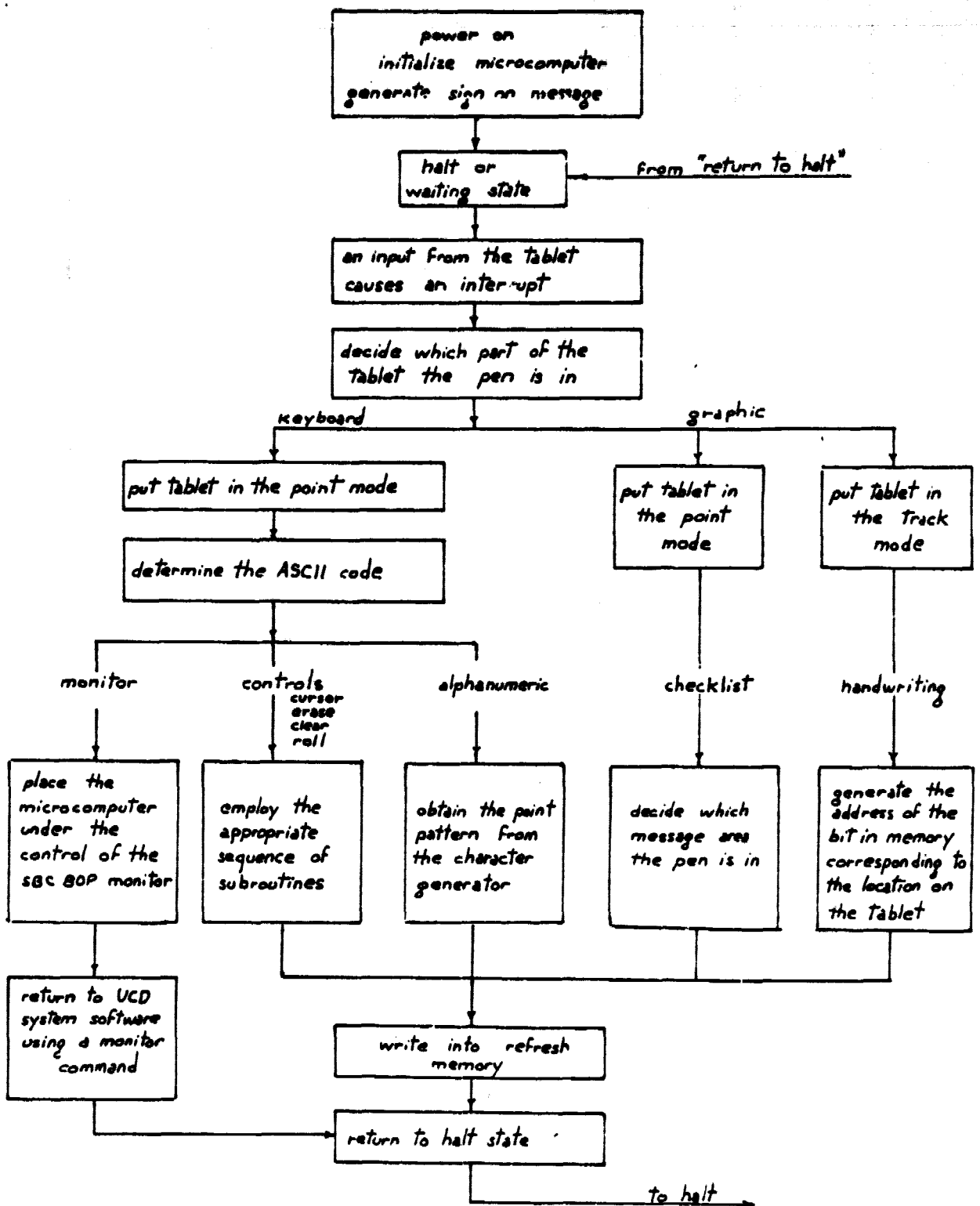


Figure 10

A flowchart of the system software



When the pen is in the keyboard area the computer places the tablet in the POINT mode of operation. In this mode the tablet produces only one pair of coordinates each time the pen is touched to the tablet. When the pen is touched to the keyboard area the program follows one of many sequences of subroutines, depending on the square selected. The same procedure is followed for all of the alphanumeric characters. The ASCII code corresponding to the character is used to address the character generator and obtain the point matrix, a byte at a time. As each byte is generated it is written into the refresh memory until the entire point array for the character has been placed in the memory.

If the character is a control character the program branches to an appropriate sequence of subroutines. The cursor controls, which allow the operator to position the cursor, are among the more complicated. The address of the cursor in the character matrix is stored in an assigned place in RAM memory. When one of the cursor controls is touched this address is retrieved and modified in accordance with the control function. The old cursor is removed from the refresh memory and the new cursor written in.

Rolling the display upward or downward is accomplished by incrementing or decrementing the starting address of the display at a fixed rate. The rate is generated by a program loop within the computer. At the end of each timed interval the computer modifies the starting address which is stored both in RAM and in a buffer memory within the display interface.

Touching the ERASE square causes the computer to branch to a sequence of subroutines that make the pen act like an eraser. When the pen is placed in the handwriting area an address in the point matrix is generated from the coordinates of the tablet, in the same way as when the pen is in write mode. However, the computer supplements this point with an array of 21 points surrounding it in an approximate circle. The corresponding points in the refresh memory are then cleared.

The operator can obtain access to the SBC 80P system monitor by touching the square labelled MONITOR. The monitor is a system software package developed by Intel that gives the operator control of the microcomputer by means of a number of relatively powerful commands. The monitor provides for: the display of register contents and selected areas of memory, the input and output of hexadecimal file by means of paper tape or teletype, and the modification of contents of registers and memory. The operator can return control of the computer to the terminal's software by means of a monitor command which causes a jump to any location in memory.

The checklist mode of operation, which is intended to be the major means of communication, has only a token amount of software devoted to it. The existing software divides the checklist area into seven large sections, each with a corresponding space in RAM memory. When the pen is touched to a section the contents of the associated section of RAM is written into the refresh memory and displayed on the screen. This arrangement allows an elementary demonstration of checklist operation to be made. However, it only suggests the potential of the checklist mode. In actual practice the checklist area would be divided into small squares and information about the checklist would be stored in the central computer.

### SUGGESTIONS FOR FURTHER WORK

The terminal, while useful in itself, can be considered the first stage in the development of a user-compatible data entry system. As delivered, the terminal has a minimum software package. It also falls short of the ultimate goal of portability. The ideal terminal should be portable so that it can be carried by the user, much like a notepad. Further work can proceed along several lines; some designed to expand the software capability and some directed toward achieving a portable design.

The unique features of the terminal are its use of checklist inputs and short handwritten messages. Both concepts require further work if their potential is to be realized.

The first step in the development of the checklist mode of entry should be the design of a higher level language that allows the user to quickly and conveniently specify his own checklist. At present, tedious assembly language programming must be used. However, by using both graphical and alphanumeric input from the electronic tablet, a language can be devised that allows a user to specify a checklist with only a few instructions. Graphical input can be used to specify the format and define the spaces while alphanumeric input can be used to list the questions and responses.

Digitized handwriting is entirely different from the usual alphanumeric output of a terminal and requires unique processing, coding, and storage. Although it is known that the information content of handwriting is low, the electronic tablet generates an enormous amount of data. Therefore, an opportunity exists for efficient compression of the data. Development of these techniques is necessary before digitized handwriting can be rapidly transmitted and economically stored.

While the preceding suggestions require the development of software, the development of a portable terminal necessitates considerable hardware modification. Two main factors contributing to the bulkiness of the present tablet are: the weight of the electronic tablet and its electronics and the lack of compactness due to the fact that the terminal is assembled from an array of unrelated parts. The electronic tablet is a precision device, overdesigned for this application. A tablet designed for the needs of the terminal could be lighter, and furthermore, its electronics could be incorporated as a function of the microcomputer. Integrating the electronic tablet and the other components of the terminal would eliminate many overlapping parts that now exist. Much of the present size and weight is due to duplicate power supplies, cables, and buffering electronics.

## REFERENCES

1. A User's Guide for the UCD-NASA Terminal. E. W. Owen, C. A. James, and P. C. Jain, Department of Electrical Engineering, University of California, Davis, March 1977.
2. The Hardware Manual for the UCD-NASA Terminal. (as above)
3. SBC 80P Prototyping Package Users Guide. Intel Corporation, 3005 Bowers Avenue, Santa Clara, CA 95051, 1976.
4. SBC 80/10 Hardware Reference Manual. (as above)
5. A User-Compatible Terminal for Medical Applications. E. W. Owen and E. C. Moseley, Computers in Biology and Medicine, Vol. 7, No. 2, 1977.

## APPENDIX

SBC 80/10 Single Board Computer

SBC 80P Prototype Package

SBC 016 16 klyte RAM Memory Board

This material has been taken from the Microcomputer  
Systems Data Book, Intel Corp., 1976

## SBC 80/10 SINGLE BOARD COMPUTER

### 8080A Central Processing Unit

1 K bytes of read/write memory

Sockets for 4K bytes of programmable or masked read-only memory

48 programmable parallel I/O lines with sockets for interchangeable line drivers and terminators

Programmable Synchronous/Asynchronous communications interface with selectable teletype-writer or RS232C compatibility

Six interrupt request lines

Bus drivers for memory and I/O expansion

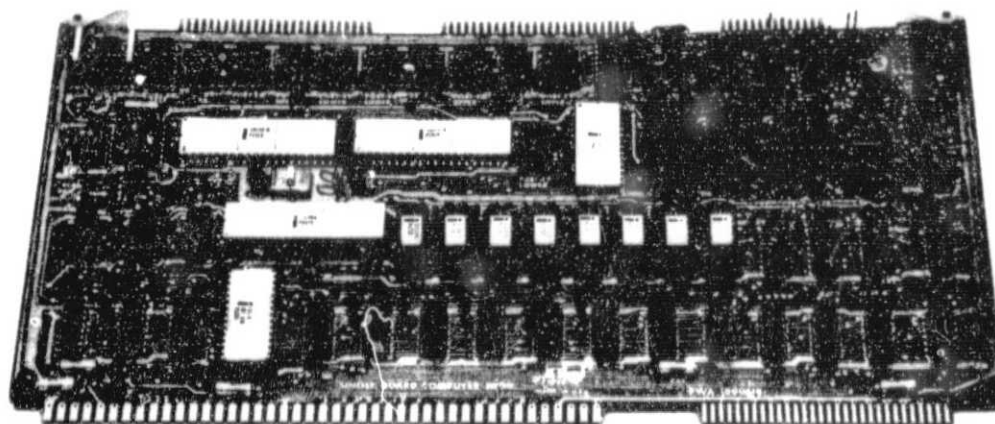
Compatible with optional memory and I/O expansion boards.

The SBC 80/10 is a member of Intel's complete line of OEM computer systems which take full advantage of Intel's LSI technology to provide economical, self-contained computer based solutions for OEM applications. The SBC 80/10 is a complete computer system on a single 6.75-by-12 inch printed circuit card. The CPU, system clock, read/write memory, non-volatile read-only-memory, I/O ports and drivers, serial communications interface, bus control logic and drivers all reside on the board.

Intel's powerful 8-bit n-channel MOS 8080A CPU, fabricated on a single LSI chip, is the central processor for the SBC 80/10. The 8080A contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs providing both single and double precision operators.

The 8080A has a 16-bit program counter which allows direct addressing of up to 64K bytes of memory. An external stack, located within any portion of read/write memory, may be used as a last in/first out stack to store the contents of the program counter, flags, accumulator and all of the six general purpose registers. A sixteen bit stack pointer controls the addressing of this external stack. This stack provides subroutine nesting that is bounded only by memory size.

The SBC 80/10 contains 1K bytes of read/write memory using Intel<sup>®</sup> 8111 low power static RAM. All on-board RAM read and write operations are performed at maximum processor speed. Sockets for up to 4K bytes of non-volatile read-only memory are provided on the board. Read only memory may be added in 1K byte increments using Intel 8708 erasable and electrically reprogrammable ROMs (EPROMs) or Intel 8308 masked ROMs. All on-board ROM read operations are performed at maximum processor speed.



The SBC 80/10 contains 48 programmable parallel I/O lines implemented using two Intel 8255 Programmable Peripheral Interfaces. The system software is used to configure the I/O lines in any combination of unidirectional input/output, and bi-directional ports indicated in Table 1. Therefore, the I/O interface may be customized to meet specified peripheral requirements. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. Hence, the flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 48 programmable I/O lines and signal ground lines are brought out to two 50 pin edge connectors that mate with flat-cable or round-cable.

A programmable communications interface using Intel's 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the board. A jumper selectable baud rate generator provides the USART with all common communications frequencies. The USART can be programmed by the system software to select the desired asynchronous or synchronous serial data transmission technique (including IBM Bi-Sync). The mode of operation (i.e. synchronous or asynchronous), data format, control character format, parity, and asynchronous transmission rate are all under program control. The 8251 provides full duplex, double buffered transmission and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The inclusion of jumper selectable TTY or RS232C compatible interfaces on the board, in conjunction with the USART provide a direct interface to teletypes, CRTs, RS232 compatible cassettes, asynchronous and synchronous modems. The RS232C or TTY command lines, serial data lines, and signal ground lines are brought out to a 25-pin edge connector that mates with RS232C compatible flat or round cable.

Interrupt requests may originate from six sources. Two jumper selectable interrupt requests can be automatically generated by the Programmable Peripheral Interface when a byte of information is ready to be transferred to the CPU (i.e. input buffer is full) or a byte of information has been transferred to a peripheral device (i.e. output buffer is

empty). Two jumper selectable interrupt requests can be automatically generated by the USART when a character is ready to be transferred to the CPU (i.e. receive channel buffer is full) or a character is ready to be transmitted (i.e. transmit channel data buffer is empty). These four interrupt request lines are all individually maskable under program control. Two interrupt request lines may be interfaced directly to user designated peripheral devices; one via the system bus and the other via the I/O edge connector. The six interrupt request lines share a single CPU interrupt level. When an interrupt request is recognized, a RESTART 7 instruction is generated. The processor responds by suspending program execution and executing a user defined interrupt service routine originating at location 38<sub>16</sub>.

SBC 80 memory and I/O capacity may be increased by adding standard Intel memory and I/O Loads. Memory may be expanded to 64K bytes by adding user specified combinations of SBC-016 16K RAM boards, SBC 416 16K PROM boards, and SBC-406 6K PROM boards. Input/output capacity may be increased to 504 input lines and 504 output lines using SBC-508 I/O boards, containing 32 input lines and 32 output lines per board. Memory and I/O may be increased simultaneously by adding an SBC-104 board containing 4K bytes of RAM, sockets for 4K bytes of PROM, 48 programmable I/O lines and a USART. Modular expandable backplanes and card cages, with a four board capacity, are available to support multi-board systems.

The development cycle of SBC 80/10 based products may be significantly reduced using the Intel<sup>®</sup> Microcomputer Development System. The resident macro-assembler, text editor, and system monitor greatly simplify the design, development, and debug of SBC 80/10 system software. An optional Diskette Operating System allows programs to be loaded, assembled, edited, and executed faster than using conventional paper tape, card, or cassette peripherals. A unique In-Circuit Emulator (ICE-80) option provides the capability of developing and debugging software directly on the SBC 80/10.

Intel's high-level programming language, PL/M, provides the capability to program in a natural, algorithmic language and eliminates the need to manage register usage or allocate memory. PL/M programs can be written in a much shorter time than assembly language programs.

TABLE 1 INPUT/OUTPUT PORT MODES OF OPERATION

PORT	NO. OF LINES	MODE OF OPERATION				BIDIRECTIONAL	CONTROL
		UNIDIRECTIONAL					
		INPUT		OUTPUT			
		UNLATCHED	LATCHED & STROBED	LATCHED	LATCHED & STROBED		
1	8	X	X	X	X	X	
2	8	X	X	X	X		
3	8	X		X			X <sup>1</sup>
4	8	X		X			
5	8	X		X			
6	4	X		X			
	4	X		X			

1. Note: Port 3 must be used as a control port when either Port 1 or Port 2 are used as a latched and strobed input or a latched and strobed output or Port 1 is used as a bidirectional port.



**SPECIFICATIONS****WORD SIZE**

Instruction: 8, 16, or 24 bits  
Data: 8 bits

**CYCLE TIME**

Basic Instruction Cycle: 1.95  $\mu$ sec  
Note: Basic instruction cycle is defined as the fastest instruction (i.e. four clock cycles)

**MEMORY ADDRESSING**

On-Board ROM/PROM: 0-0FFF  
On-Board RAM: 3C00-3FFF

**MEMORY CAPACITY**

On-Board ROM/PROM: 4K bytes (sockets only)  
On-Board RAM: 1K bytes  
Off-Board Expansion: Up to 65,536 bytes using user specified combinations of RAM, ROM, and PROM

Note: ROM/PROM may be added in 1K byte increments.

**I/O ADDRESSING**

On-Board Programmable I/O (See Table 1)

Port	8255 No. 1						8255 No. 2		USART Data	USART Control
	1	2	3	4	5	6	No. 1 Control	No. 2 Control		
Address	E4	E5	E6	E8	E9	EA	E7	EB	EC	ED

**I/O CAPACITY**

Parallel: 48 programmable lines (See Table 1)

Note: Expansion to 504 input and 504 output lines can be accomplished using optional I/O boards.

**SERIAL BAUD RATES**

Frequency (KHz) (Jumper Selectable)	Baud Rate (Hz)	
	Synchronous	Asynchronous (Program Selectable)
307.2		16 64
153.6		19200 4800
76.8		9600 2400
38.4		4800 1200
19.2	38400	2400 600
9.6	19200	1200 300
4.8	9600	600 150
2.4	4800	300 75
1.2	6980	110

**SERIAL COMMUNICATIONS CHARACTERISTICS****Synchronous:**

5-8 bit characters  
Internal or external character synchronization  
Automatic Sync Insertion

**Asynchronous:**

5-8 bit characters  
Break character generation  
1, 1-1/2, or 2 stop bits  
False start bit detectors

**INTERRUPTS**

Single-level with on-board logic that automatically vectors processor to location 3816 using RESTART 7 instruction. Interrupt requests may originate from user specified I/O (2) the programmable peripheral interface (2), or USART (2).

**INTERFACES**

Bus: All signals TTL compatible  
Parallel I/O: All signals TTL compatible  
Serial I/O: RS232C, or a 20 mil current loop TTY interface (jumper selectable)  
Interrupt Requests: All TTL compatible (active low)

**SYSTEM CLOCK**

2.048 MHz  $\pm 0.1\%$

**CONNECTORS**

Interface	No. of Double-Sided Pins	Centers (in.)	Mating Connectors
Bus	86	0.156	CDC VP801E43D00A1 Wire-Wrap
Parallel I/O (2)	50	0.1	3M 3415-000 Flat Pins or TI H312125 Flat Pins
Serial I/O	26	0.1	3M 3462-0001 Flat or AMP 88106-1 Flat

**PHYSICAL CHARACTERISTICS**

Width: 12.00 in. (30.48 cm) Depth: 0.50 in. (1.27 cm)  
Height: 6.75 in. (17.15 cm) Weight: 14 oz. (484.4 gm)

**ELECTRICAL CHARACTERISTICS**

DC Power:	w/o PROM <sup>1</sup>	with PROM <sup>2</sup>
V <sub>CC</sub> = +5 $\pm 5\%$	I <sub>CC</sub> = 2.9A max	4.0A
V <sub>DD</sub> = +12 $\pm 5\%$	I <sub>DD</sub> = 140 mA max	400 mA
V <sub>BB</sub> = -5V $\pm 5\%$	I <sub>BB</sub> = 2 mA max	200 mA
V <sub>AA</sub> = -12V $\pm 5\%$	I <sub>AA</sub> = 175 mA max	175 mA

Notes: 1. Does not include power required for optional PROM, I/O drivers, and I/O terminators.

2. With four 2708 PROMs and ten 220/330 $\Omega$  resistor packs installed, all low.

**LINE DRIVERS AND TERMINATORS****I/O Drivers:**

The following line drivers and terminators are all compatible with the I/O driver sockets on the SBC 80/10.

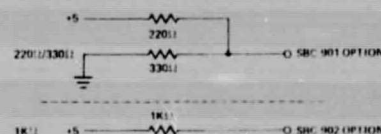
Driver	Characteristic	Sink Current (mA)
7438	I, OC	48
7437	I	48
7432	NI	16
7426	I, OC	16
7409	NI, OC	16
7408	NI	16
7403	I, OC	16
7400	I	16

Note: I = inverting, NI = non-inverting, OC = open collector

Port 1 has 25 nA totem pole drivers and 1 k $\Omega$  terminators.

**I/O Terminators:**

Terminators: 220 $\Omega$ /330 $\Omega$  divider or 1 k $\Omega$  pull up

**Bus Drivers:**

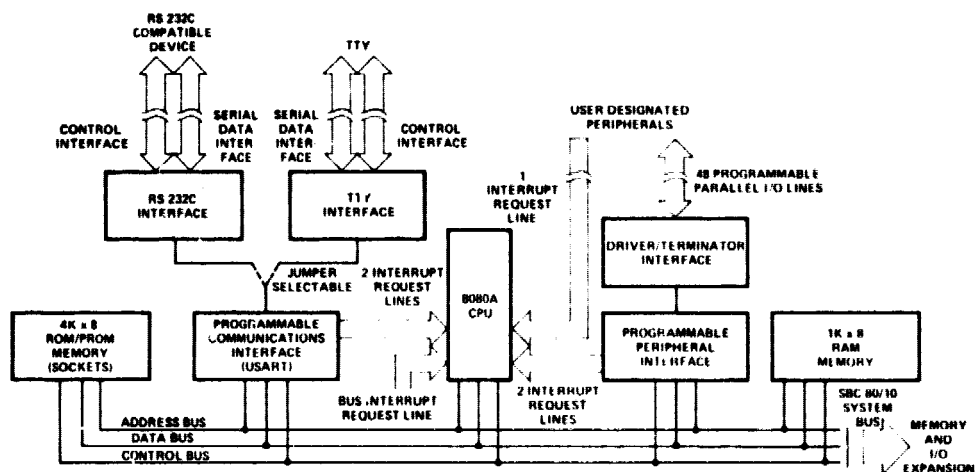
Function	Characteristic	Sink Current (mA)
Data	T <sub>H</sub> State	25
Address	T <sub>H</sub> State	25
Commands	T <sub>H</sub> State	25

**ENVIRONMENTAL**

Operating Temperature: 0°C to 55°C

**COMPATIBLE BOARDS**

SBC-016 16K byte RAM  
SBC-406 6K byte PROM  
SBC-416 16K byte PROM  
SBC-508 32 input lines/32 output lines  
SBC-104 4K byte RAM, 4K byte PROM,  
48 prog. I/O lines, USART



1. Interrupts originating from the Programmable Communications Interface and Programmable Peripheral Interface are jumper selectable.

**SBC 80/10 BLOCK DIAGRAM**

## SBC 80P PROTOTYPE PACKAGE

SBC 80/10 Single Board Computer

SBC 604 Cardcage/Backplane with compatible power supply cables

Comprehensive System Monitor residing on two Intel® 8708 EPROMs

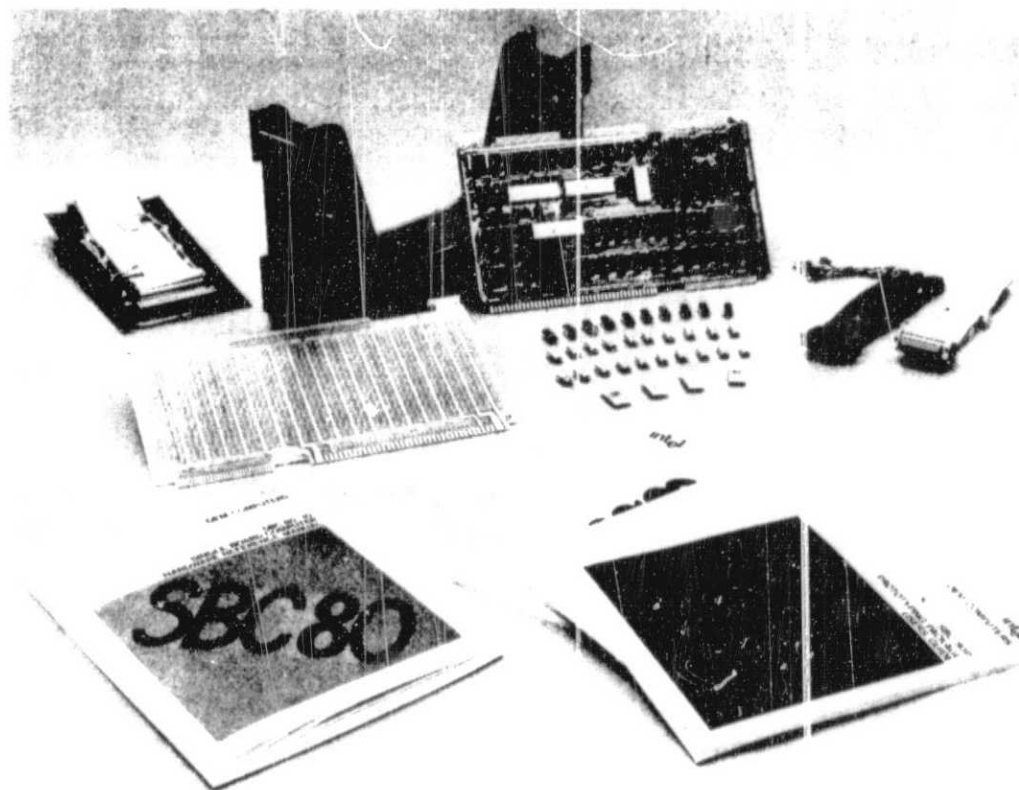
RS232C and TTY cables to interface the SBC 80/10 to any RS232C compatible device or teletypewriter

Two 50-pin unterminated flat cables with connectors that mate with SBC 80/10 parallel I/O PC edge connectors

Full complement of EPROMs, I/O line drivers, and I/O line terminators

SBC 905 Universal Prototype Board for interfacing custom hardware to the SBC 80/10

The SBC 80P Prototype Package contains all the hardware, software, and documentation necessary to evaluate Intel's SBC 80/10 Single Board Computer for OEM applications.



The heart of the SBC 80P Prototype Package is the SBC 80/10 Single Board Computer, a complete computer on a single 6.75-by-12 inch printed circuit board. The SBC 80/10 includes an 8080A CPU, 1K bytes of RAM memory, sockets for 4K bytes of EPROM memory, 48 programmable parallel I/O lines with sockets for interchangeable line drivers and terminators, a programmable synchronous/asynchronous communications interface with RS232C and teletype compatibility, a multi-source single level interrupt network and bus drivers for memory and I/O expansion.

An SBC 604 Modular Cardcage/Backplane is included to house the SBC 80/10 and provide an easily accessible bus interface. The SBC 604 houses the SBC 80/10 and up to three expansion boards. All SBC 80 bus signals are present on all four mating connectors. Also included are two power supply cables, which mate with the power supply connectors on the backplane, to carry  $\pm 5$  and  $\pm 12$  volts DC.

A comprehensive system monitor, residing in two Intel® 8708 EPROMs, is included to facilitate the loading, execution, and debug of SBC 80/10 based programs. Monitor commands include the ability to read and write hexadecimal paper tapes, execute pre-defined program segments, display and alter memory contents, display and alter CPU register contents. Monitor commands and resulting information may be initiated and displayed using a teletype or CRT terminal. Two cables are provided for this purpose. The first interconnects the serial PC edge connector on the SBC 80/10 to any RS232C compatible device. The second connects the RS232C cable to a teletype.

Wire-wrap jumpers on the SBC 80/10 select either teletype or RS232C operation and a jumper selectable baud rate

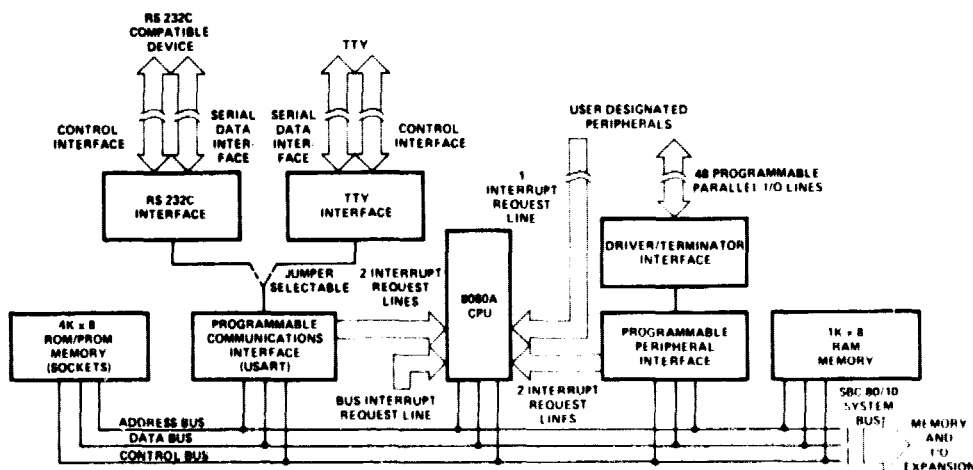
generator on the SBC 80/10 is used to select the appropriate communications frequency.

Two 50-pin unterminated flat cables are included to facilitate interfacing the 48 parallel I/O lines on the SBC 80/10 to user designated I/O devices. The 48 programmable I/O signal lines and corresponding 48 ground lines on the SBC 80/10 are brought out to two 50-pin PC edge connectors where they mate with the two flat cables. The cables are left unterminated to allow the user to provide the appropriate mating connector for any application.

The SBC 80P prototype package includes a full complement of EPROMs, I/O line drivers, and I/O line terminators. Four Intel 8708 EPROMs (1K bytes each) are included. Two EPROMs contain the system monitor and two are unprogrammed. Ten 7437 48-milliamp TTL quad I/O line drivers, ten Intel SBC 901 220 $\Omega$ /330 $\Omega$  line terminators, and ten SBC 902 1 k $\Omega$  line terminators are included.

An SBC 905 Universal Prototype Board is provided to facilitate the construction of SBC 80/10 customized I/O and/or memory hardware. The SBC 905 plugs directly into the SBC 604 Cardcage/Backplane and can house up to 95 16-pin wire-wrap sockets or an equivalent mix of 14, 16, 18, 22, 24, 28, or 40-pin sockets.

The SBC 80P Prototype contains all the in-depth documentation needed to program and interface the SBC 80/10 Single Board Computer. An 8080 Assembly Language Manual, PL/M Programming Manual, SBC 80/10 Hardware Reference Manual, and SBC 80P User's Guide are all included to provide clear and concise information relevant to the use of the SBC 80/10 in OEM equipment.



1. Interrupts originating from the Programmable Communications Interface and Programmable Peripheral Interface are jumper selectable.

SBC 80/10 BLOCK DIAGRAM

**SPECIFICATIONS****SINGLE BOARD COMPUTER**

- (1) SBC 80/10 Single Board Computer

**CARDCAGE/BACKPLANE**

- (1) SBC 604 Modular Cardcage/Backplane with capacity for four SBC boards.

**CABLES**

- (2) Power Supply Cables (2 ft long):  
Both required for  $\pm 5$ ,  $\pm 12$  volts DC; mate with SBC 604.
- (2) 50-wire parallel I/O Flat Cables (5 ft long):  
Both mate with SBC 80/10 50-pin parallel I/O PC edge connectors.
- (1) RS232C Cable (2 ft long):  
Flat cable with 26-pin SBC 80/10 connector on one end and a standard 25-pin RS232C connector on the other end.
- (1) TTY Cable (5 ft long):  
Interconnects RS232C cable with teletype; 25-pin RS232C mating connector on one side; seven spade lugs on the other end.

**I/O LINE DRIVERS AND TERMINATORS**

- (10) 7437 48 mA open collector line drivers
- (10) SBC 901 220 $\Omega$ /330 $\Omega$  terminators
- (10) SBC 902 1 k $\Omega$  terminators

**UNIVERSAL PROTOTYPE BOARD**

- (1) SBC 905 Universal Prototype Board with capacity for 95 16-pin wire-wrap sockets or equivalent mix of 14, 16, 18, 22, 24, 28, or 40-pin sockets; compatible with SBC 604 cardcage/backplane.

**SYSTEM MONITOR****Addresses:**

0000–08FF<sub>H</sub> (ROM; 3F80–3FFF<sub>H</sub> (RAM)

**Commands:**

Display Memory (D)  
Program Execute (G)  
Insert Instructions into Memory (I)  
Move Memory (M)  
Read Hexadecimal File (R)  
Substitute Memory (S)  
Write Hexadecimal File (W)  
Examine and Modify CPU Registers (X)

**Drivers:**

Console Input  
Console Output  
Reader Input  
Punch Output

**Breakpoints:**

A hardware breakpoint capability may be implemented by an interrupt service routine beginning at RAM location 3C3D<sub>H</sub>. Typically, a 2-byte call is used. Interrupt generation at the breakpoint may be accomplished by user hardware or by insertion of single byte calls at instruction boundaries.

**LITERATURE**

8080 Assembly Language Manual  
PL/M Programming Manual  
SBC 80/10 Hardware Reference Manual  
SBC 80P User's Guide  
SBC 80/10 Schematics

## SBC-016 16K BYTE RAM MEMORY BOARD

**SBC-80 RAM memory expansion through direct bus interface**

**16K byte read/write memory capacity**

**On-board hardware for refresh of all dynamic memory elements**

**Jumper selectable starting address for 16K contiguous addresses**

**Read/write data buffers**

**TTL compatible data, address, and command signal interface**

The SBC-016 is a member of Intel's complete line of SBC-80 memory and I/O expansion boards. The SBC-016 interfaces directly to any SBC-80 single board computer, via the system bus, to expand RAM memory capacity.

The board contains 16K bytes of read/write memory, implemented using 32 Intel® 2107 dynamic RAM memory components. On-board refresh hardware refreshes 64 bit positions of all 32 RAM elements every 1.0 milliseconds. Each refresh cycle utilizes memory for 735 nanoseconds. If a read or write cycle is in progress when a refresh cycle is scheduled to begin, the refresh cycle is postponed until the end of the read or write cycle.

The SBC-016 contains a jumper that is used to select contiguous 16K byte address segments that begin in location 0000, 4000, 8000, or C000.

Read/write buffers reside on the board to buffer all data that is written into or read from the memory array. All data, address, and command signals on the bus interface are TTL compatible.

